

Jeu Du Taquin :

Noah Djafer
Joffrey Fouche
Groupe 4

- **Objectifs Du Devoir :**

La finalité du projet était de réaliser un jeu du taquin. C'est un jeu qui consiste, ici, à réorganiser une image qui a été mélangé afin de retrouver le pattern de base.

- **Fonctionnalités :**

Notre jeu propose trois photos différentes, chacune d'une taille différente.



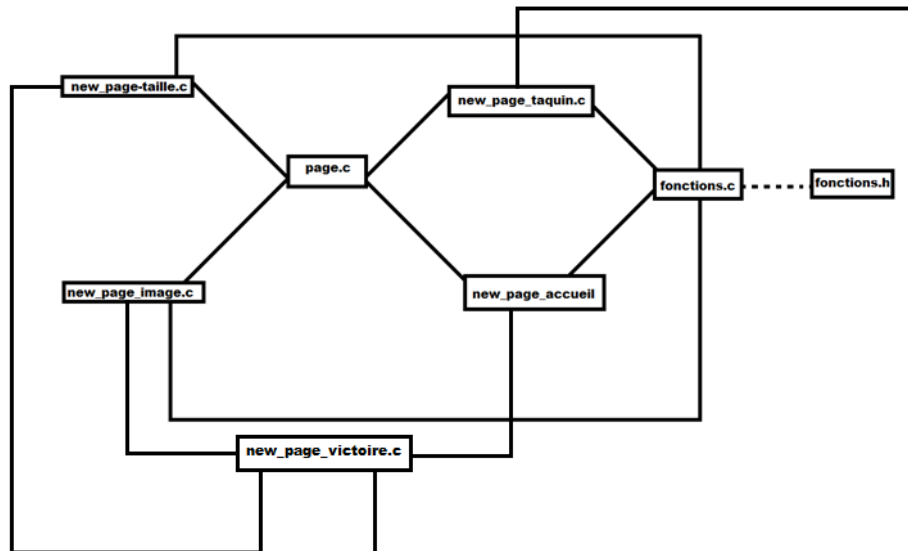
Il permet également de choisir le nombre de lignes et de colonnes que le joueur voudra pour démarrer sa partie.



- **Structure :**

Notre programme est découpé en plusieurs fichiers :

- Le fichier central qui connecte toutes les autres pages s'appelle *page.c* . Il permet de faire les transitions dans un ordre bien précis et de revenir à un menu précédent selon l'envie du joueur.
- Un fichier *fonction.c* qui contient les fonctions dont on a eu besoin tout au long du programme ainsi que son fichier d'en-tête.
 - Le reste des fichiers correspondent chacun à une page du jeu :
new_page_accueil.c / *new_page_image.c* / *new_page_taille.c* / *new_page-taquin.c* / *new_page_victoire.c*
- Finalement, Un fichier *Makefile* pour permettre une compilation rapide.



● Avancement de la partie :

Pour mesurer l'avancement d'une partie, on a choisi d'associer chaque case du taquin à une case d'un tableau en deux **dimensions égales** à celles choisies sur la page **taille**.

Le tableau est remplie de 0 à (nombre de lignes * nombre de colonnes) -1, par exemple pour un taquin 3*3 : de 0 à 8. La case du 0 correspond à la case vide du taquin.

Lorsque l'on interchange deux cases, les valeurs du tableau changent en respectant les déplacements de la case qui leurs est associées.

Finalement pour remporter la partie, il faut que les valeurs du tableau soient remis dans le bon ordre ce qui déclenchera la victoire.

● Vérification :

Pour vérifier que le taquin n'est pas impossible il faut comparer la parité du nombre de coups pour remettre zéro a la position initiale à la parité du nombre de permutation de nombre pour que le taquin soit résolu. On commence par faire une

boucle while dans le taquin numérique qui ne s'arrête que lorsque le taquin devient résoluble. On crée un deuxième tableau identique au premier pour pouvoir le modifier. On compte ensuite le nombre de coût de la case 0 pour qu'elle passe en position (0,0), on trouve ensuite la parité de ce nombre. On cherche si chaque élément du tableau est identique en termes de contenu et de place dans le tableau. Si ce n'est pas le cas on permute le nombre de notre case avec le bon nombre dans une autre case. On compte également le nombre de mouvements que l'on a fait. Et on définit la parité de ce nombre. Si les deux nombres ont la même parité. On sort et le taquin est résoluble. Sinon on recommence.

● Conclusion :

Noah : Personnellement, j'ai trouvé cet exercice plus compliqué que la première SAE que l'on a dû réaliser précédemment - *qui l'eut cru ?* -Cependant, je l'ai quand même trouvé utile et divertissant :

- Utile car j'ai pu me rendre compte des notions de C que je ne maîtrisais pas et presque les surmonter - *c'est déjà pas mal :-)*. Je me suis même rendu compte que je résistais plutôt bien à la pression contrairement à mon binôme °*° (ne m'en veut pas trop quand tu liras ceci).
- Divertissant, parce qu'on a quand même passé des bons moments dans ces dizaines d'heures en salle machine. Fin, j'ai une nouvelle addiction aux chocolats du CROUS à régler mais... un problème à la fois.

Je reste frustré malgré tout car je n'ai pas réussi à résoudre un seul de nos **propres** taquins...

Joffrey : Pour conclure, j'ai trouvé cette SAE extrêmement difficile. Elle nous a demandé un temps de travail et une concentration colossale. Chaque étape était divisé en sous étapes qui elles-mêmes étaient divisés en sous étapes. A chaque fois que l'on avançait on se rendait compte qu'il nous restait encore toute la SAE à faire. Cependant cela nous appris à diviser à scinder notre travail. On a fait beaucoup d'erreurs, on a corrigé grand nombre de ces erreurs mais il nous en restait encore plein. Comme nous l'avons poussé dans notre Git c'était une véritable chute aux enfers, et nous avons failli abandonner. Mais à force d'effort et de travail on a fini par y arriver.